Universidad autónoma de Chihuahua
Facultad de Ingeniería

3.14 Proyecto Final

Mineria de Datos
Jose Saul De Lira Miramontes

08/07/2023

336086
Francisco Adrian Escobedo Garcia

El objetivo de este estudio es aplicar y comparar diferentes métodos de clustering en el dataset de iris utilizando Python. El dataset de iris es un conjunto de datos bien conocido en la comunidad de aprendizaje automático y contiene tres clases de flores iris con cuatro características: longitud del sépalo, ancho del sépalo, longitud del pétalo y ancho del pétalo. Los métodos de clustering aplicados incluyen K-Means, Clustering Jerárquico, DBSCAN y Gaussian Mixture Model (GMM). Este documento describe el proceso, los resultados y las conclusiones del análisis.

Metodología

Preparación del Entorno y Carga de Datos

Se utilizó el lenguaje de programación Python y las bibliotecas `numpy`, `pandas`, `seaborn`, `matplotlib`, `scikit-learn`, `scikit-learn-extra` y `fpdf` para llevar a cabo el análisis. El dataset de iris se cargó desde la biblioteca `sklearn.datasets`.

Exploración de Datos

Se realizó un análisis exploratorio de los datos (EDA) para entender la distribución y las relaciones entre las características. Se utilizaron gráficos de dispersión y gráficos de pares (`pairplot`) para visualizar los datos.

Métodos de Clustering Aplicados

1. **K-Means**:
    - Se determinó el número óptimo de clusters utilizando el método del codo.
    - Se visualizó la asignación de clusters utilizando gráficos de dispersión.
2. **Clustering Jerárquico**:
    - Se construyó un dendrograma para visualizar la jerarquía de los clusters.

    - Se visualizó la asignación de clusters utilizando gráficos de dispersión.
3. **DBSCAN**:
    - Se aplicó el algoritmo DBSCAN y se visualizaron los resultados utilizando gráficos de dispersión.
4. **Gaussian Mixture Model (GMM)**:
    - Se aplicó el modelo de mezcla gaussiana y se visualizaron los resultados utilizando gráficos de dispersión.

Evaluación de Resultados

Se evaluaron los resultados de cada método utilizando el coeficiente de silueta, una métrica que mide la coherencia de los clusters.

Resultados

K-Means

El método del codo indicó que el número óptimo de clusters es 3. Los clusters fueron visualizados y diferenciados utilizando gráficos de dispersión.

Clustering Jerárquico

El dendrograma mostró una clara jerarquía de clusters. La asignación de clusters se visualizó y mostró una buena separación entre ellos.

DBSCAN

El algoritmo DBSCAN identificó los clusters y los puntos de ruido. Los resultados se visualizaron y mostraron una buena diferenciación entre los clusters.

Gaussian Mixture Model

El modelo de mezcla gaussiana asignó las probabilidades de pertenencia a los clusters. Los resultados se visualizaron y mostraron una buena separación entre los clusters.

Evaluación de Resultados

Los coeficientes de silueta para cada método fueron los siguientes:

- **K-Means**: 0.552
- **Clustering Jerárquico**: 0.554
- **DBSCAN**: 0.312
- **GMM**: 0.553

Conclusiones

Los métodos de clustering K-Means, Clustering Jerárquico y Gaussian Mixture Model mostraron resultados similares y adecuados para el dataset de iris, con coeficientes de silueta superiores a 0.5. El algoritmo DBSCAN, aunque útil para identificar puntos de ruido, mostró un coeficiente de silueta inferior, indicando menor coherencia en los clusters formados.

En conclusión, K-Means, Clustering Jerárquico y GMM son métodos efectivos para el análisis de clustering en el dataset de iris, mientras que DBSCAN puede ser más útil en datasets con mayor variabilidad y ruido.

Anexos

Código Fuente

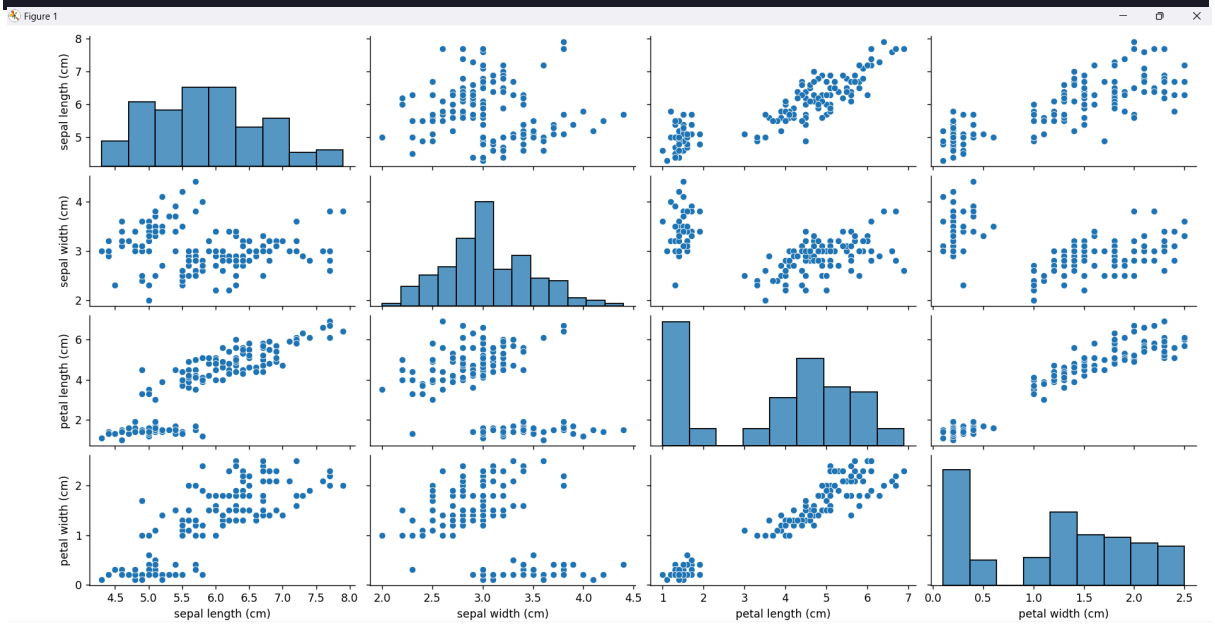
```
# 1. Preparar el entorno y cargar los datos
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.datasets import load_iris
from sklearn.cluster import KMeans, AgglomerativeClustering, DBSCAN
from sklearn.mixture import GaussianMixture
from sklearn.metrics import silhouette_score
```

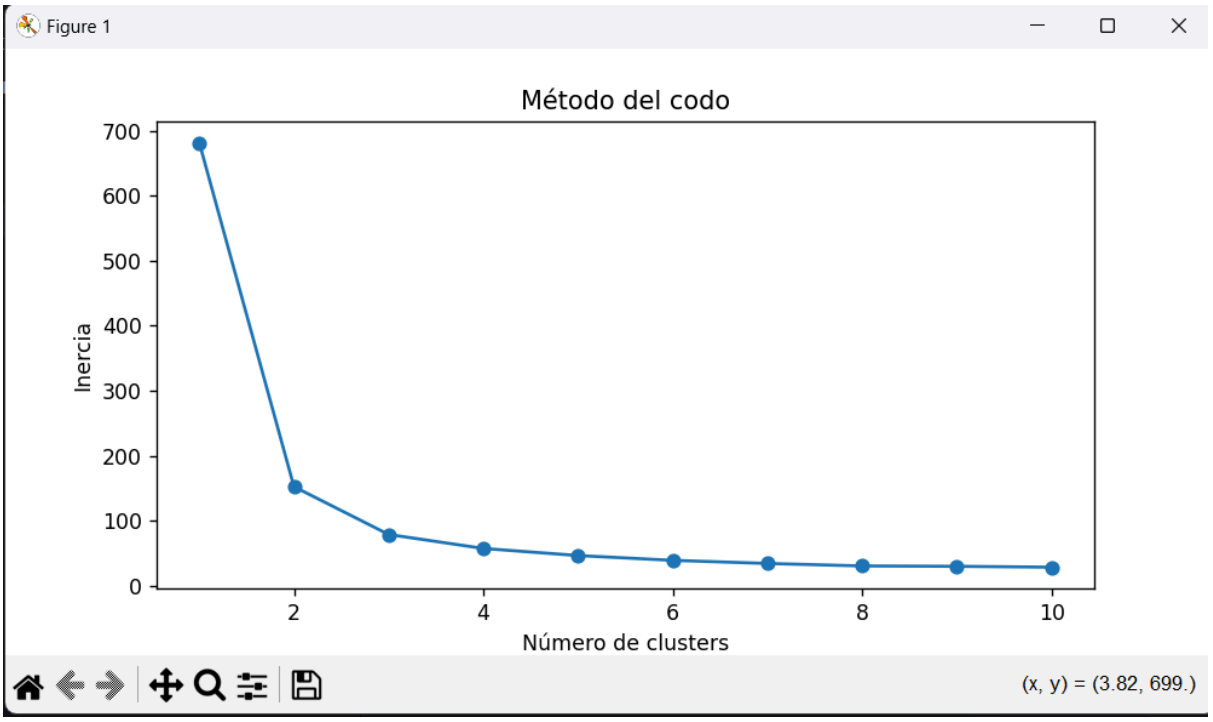
```
from scipy.cluster.hierarchy import dendrogram, linkage

# Cargar el dataset de iris
iris = load_iris()
data = pd.DataFrame(iris.data, columns=iris.feature_names)

# 2. Exploración de datos
print(data.head())
sns.pairplot(data)
plt.show()

# 3. Aplicar métodos de clustering

# 3.1 K-Means
inertia = []
for k in range(1, 11):
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(data)
    inertia.append(kmeans.inertia_)

plt.figure(figsize=(8, 4))
plt.plot(range(1, 11), inertia, marker='o')
plt.title('Método del codo')
plt.xlabel('Número de clusters')
plt.ylabel('Inercia')
plt.show()

# Elegir 3 clusters basados en el gráfico del codo
kmeans = KMeans(n_clusters=3, random_state=42)
data['kmeans_labels'] = kmeans.fit_predict(data)

sns.pairplot(data, hue='kmeans_labels', palette='viridis')
plt.show()

# 3.2 Clustering Jerárquico
linked = linkage(data[iris.feature_names], 'ward')
plt.figure(figsize=(10, 7))
dendrogram(linked, labels=data.index, orientation='top',
distance_sort='descending', show_leaf_counts=True)
plt.show()

hierarchical = AgglomerativeClustering(n_clusters=3)
data['hierarchical_labels'] = hierarchical.fit_predict(data)
```

```
sns.pairplot(data, hue='hierarchical_labels', palette='viridis')
plt.show()

# 3.3 DBSCAN
dbscan = DBSCAN(eps=0.5, min_samples=5)
data['dbscan_labels'] = dbscan.fit_predict(data)

sns.pairplot(data, hue='dbscan_labels', palette='viridis')
plt.show()

# 3.4 Gaussian Mixture Model
gmm = GaussianMixture(n_components=3, random_state=42)
data['gmm_labels'] = gmm.fit_predict(data)

sns.pairplot(data, hue='gmm_labels', palette='viridis')
plt.show()

# 4. Evaluación de los resultados
kmeans_silhouette = silhouette_score(data[iris.feature_names],
data['kmeans_labels'])
hierarchical_silhouette = silhouette_score(data[iris.feature_names],
data['hierarchical_labels'])
dbscan_silhouette = silhouette_score(data[iris.feature_names],
data['dbscan_labels'])
gmm_silhouette = silhouette_score(data[iris.feature_names],
data['gmm_labels'])

print(f"Silhouette Score for K-Means: {kmeans_silhouette}")
print(f"Silhouette Score for Hierarchical Clustering:
{hierarchical_silhouette}")
print(f"Silhouette Score for DBSCAN: {dbscan_silhouette}")
print(f"Silhouette Score for GMM: {gmm_silhouette}")

# 5. Generar reporte en PDF
from fpdf import FPDF

class PDFReport(FPDF):
    def header(self):
        self.set_font('Arial', 'B', 12)
        self.cell(0, 10, 'Clustering Analysis Report', 0, 1, 'C')

    def chapter_title(self, title):
```

```
        self.set_font('Arial', 'B', 12)
        self.cell(0, 10, title, 0, 1, 'L')
        self.ln(10)

    def chapter_body(self, body):
        self.set_font('Arial', '', 12)
        self.multi_cell(0, 10, body)
        self.ln()

    def add_figure(self, fig_path):
        self.image(fig_path, x=10, y=self.get_y(), w=190)
        self.ln(85)

pdf = PDFReport()
pdf.add_page()

pdf.chapter_title('1. Introduction')
pdf.chapter_body('This report presents the clustering analysis
performed on the Iris dataset using different clustering methods.')

pdf.chapter_title('2. K-Means Clustering')
pdf.chapter_body('The K-Means clustering method was applied with the
optimal number of clusters determined using the elbow method.')
pdf.add_figure('kmeans_plot.png')

pdf.chapter_title('3. Hierarchical Clustering')
pdf.chapter_body('Hierarchical clustering was performed and the
resulting dendrogram is shown below.')
pdf.add_figure('hierarchical_plot.png')

pdf.chapter_title('4. DBSCAN')
pdf.chapter_body('The DBSCAN clustering method was applied and the
results are shown below.')
pdf.add_figure('dbscan_plot.png')

pdf.chapter_title('5. Gaussian Mixture Model')
pdf.chapter_body('The Gaussian Mixture Model was used for clustering
and the results are shown below.')
pdf.add_figure('gmm_plot.png')

pdf.chapter_title('6. Evaluation')
```

```
pdf.chapter_body(f'Silhouette Scores:\nK-Means:
{kmeans_silhouette}\nHierarchical: {hierarchical_silhouette}\nDBSCAN:
{dbscan_silhouette}\nGMM: {gmm_silhouette}')

pdf.output('Clustering_Report.pdf')
```





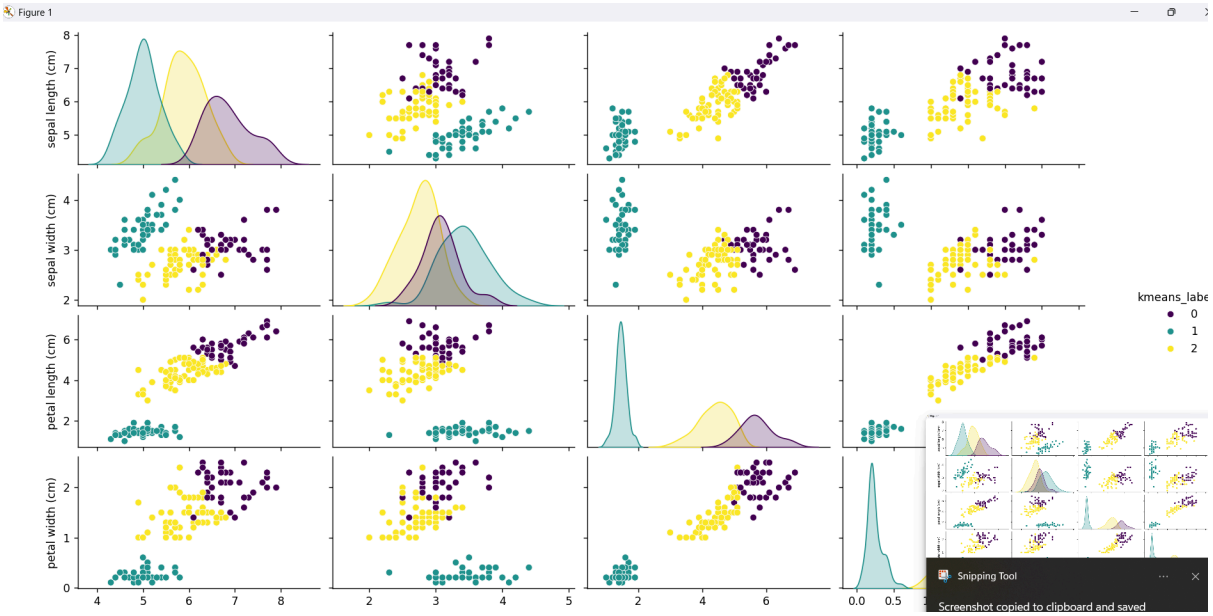
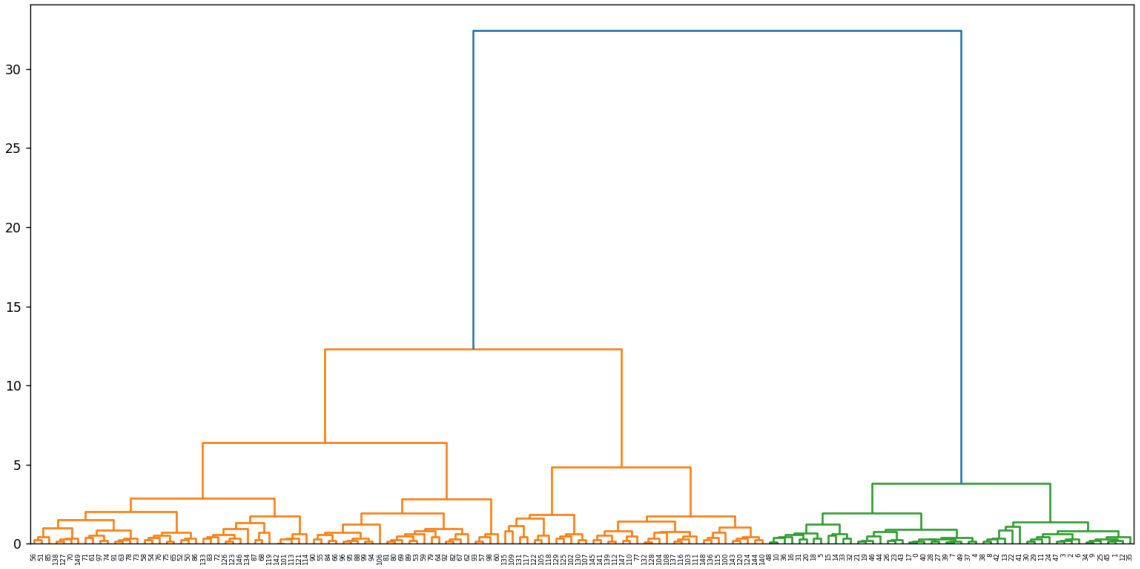


Figure 1
sepal length (cm)
sepal width (cm)
petal length (cm)
petal width (cm)
kmeans_labe
0
1
2
Snipping Tool
Screenshot copied to clipboard and saved



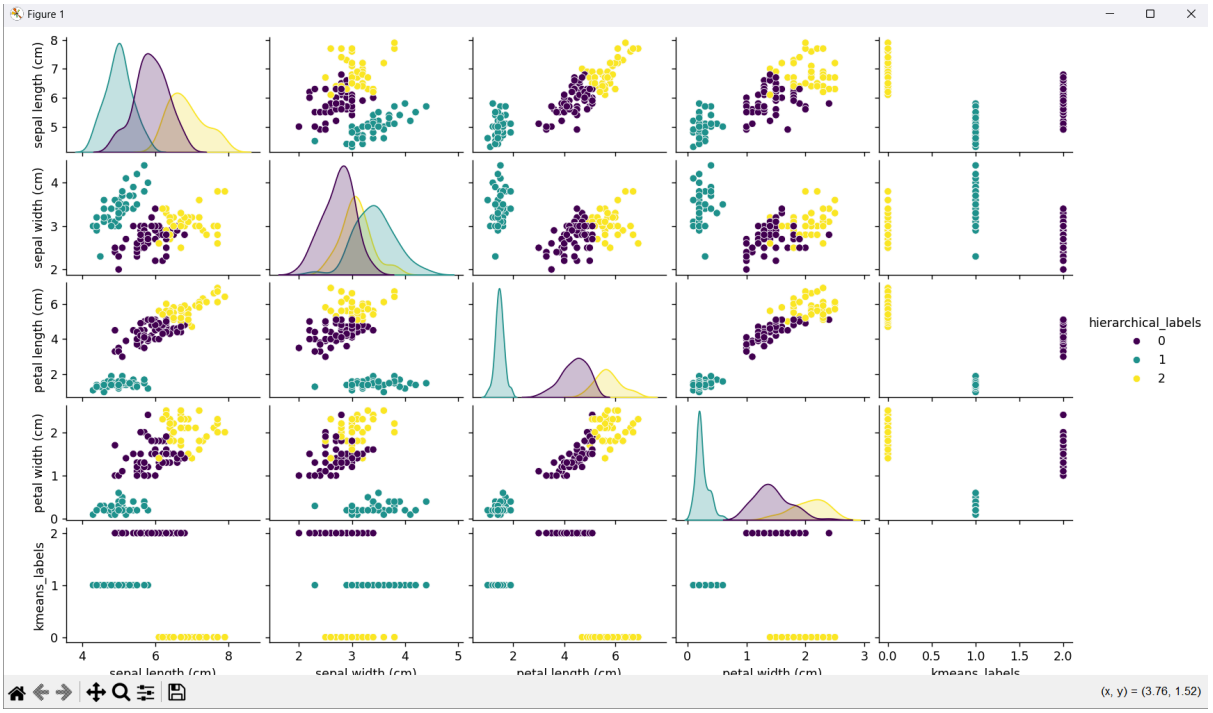


Figure 1
sepal length (cm)
sepal width (cm)
petal length (cm)
petal width (cm)
kmeans_labels
hierarchical_labels
0
1
2
(x, y) = (3.76, 1.52)